

Workflow in a paperless office

Patent Number: ☐ US2002032596
Publication date: 2002-03-14
Inventor(s): OHSAKI HIROYASU (JP); TSUDA KAZUYUKI (JP)
Applicant(s): IBM (US)
Requested Patent: ☐ JP2002099686
Application Number: US20010951096 20010912
Priority Number(s): JP20000276954 20000912
IPC Classification: G06F17/60
EC Classification: G06F17/60A
Equivalents:

Abstract

A workflow system for a paperless office, an information processing apparatus, a method for simply defining a complicated workflow, for example, a workflow such as a circulation among a plurality of departments where the circulation route varies in each department, and a storage medium. A workflow system comprises: a manipulating computer terminal for executing a workflow between persons in charge, a computer terminal for designing the workflow by designating project variables for multiplexing a plurality of paths for nodes, each indicating a unit of operation to be handled; and a workflow server for managing the designed workflow and accessing the manipulating computer terminals in accordance with activities that indicate operations assigned to the nodes

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-99686
(P2002-99686A)

(43) 公開日 平成14年4月5日 (2002.4.5)

(51) Int.Cl.⁷

G 0 6 F 17/60

識別記号

1 6 2

F I

G 0 6 F 17/60

特コード* (参考)

1 6 2 C 5 B 0 4 9

審査請求 有 請求項の数20 OL (全 22 頁)

(21) 出願番号 特願2000-276954 (P2000-276954)

(22) 出願日 平成12年9月12日 (2000.9.12)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州

アーモンク (番地なし)

(72) 発明者 大崎 博晴

神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内

(74) 代理人 100086243

弁理士 坂口 博 (外4名)

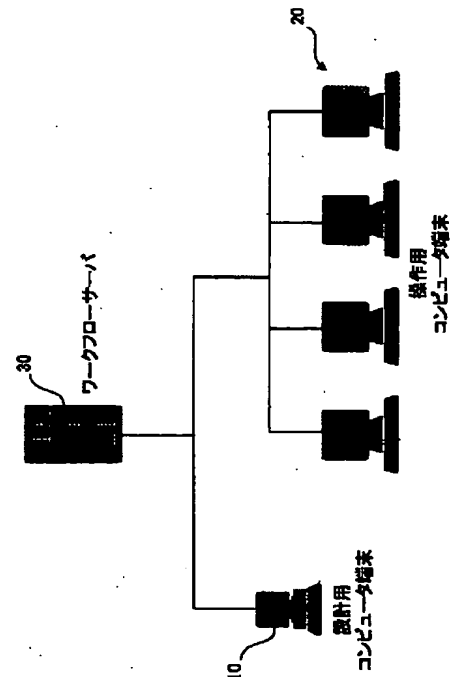
最終頁に続く

(54) 【発明の名称】 ワークフローシステム、情報処理装置、ワークフロー定義方法、記憶媒体、およびプログラム伝送装置

(57) 【要約】

【課題】 例えば回覧経路が部門毎に異なる複数部門の回覧等、複雑なワークフローを単純な定義によって実現する。

【解決手段】 ネットワークに接続され、担当者間でワークフローを実行するための操作用コンピュータ端末20と、扱われる処理の単位であるノードに対して複数の経路を多重化するためのプロジェクト変数を指定してワークフローを設計する設計用コンピュータ端末10と、設計されたワークフローを管理すると共に、ノードに割り当てられた処理であるアクティビティに対応して操作用コンピュータ端末20にアクセスするワークフローサーバ30とを備えたワークフローシステム。



【特許請求の範囲】

【請求項1】 複数のコンピュータ端末が接続されるネットワークにおいて、ワークフローに従って当該コンピュータ端末を利用する担当者間で当該ネットワークを介したデータの受け渡しを行うワークフローシステムであって、

前記担当者に割り当てられる処理であるノードと当該ノードの処理順を決定するためのパスとを定めたワークフロー定義および当該ワークフロー定義で参照されるデータ定義からなるプロジェクトを作成して前記ワークフローを設計する設計手段と、

前記設計手段により作成された前記プロジェクトを記憶する記憶手段と、

前記記憶手段に記憶された前記プロジェクトを用いて前記ワークフローの業務単位であるプロセスを作成して前記担当者に対して処理を割り当てる管理手段と、を備えたことを特徴とするワークフローシステム。

【請求項2】 前記プロジェクトにおける前記データ定義では、データとしてプロジェクト型およびその配列が定義されることを特徴とする請求項1記載のワークフローシステム。

【請求項3】 前記設計手段は、配列の個々の要素が異なる種類のプロジェクトを扱って前記ワークフローを設計することを特徴とする請求項1記載のワークフローシステム。

【請求項4】 ネットワークに接続され、担当者の間でワークフローを実行するための操作用コンピュータ端末と、

扱われる処理の単位であるノードに対して複数のプロセスを多重化するためのプロジェクト変数を指定して前記ワークフローを設計する設計用コンピュータ端末と、前記設計用コンピュータ端末により設計された前記ワークフローを管理すると共に、ノードに割り当てられた処理であるアクティビティに対応して前記操作用コンピュータ端末にアクセスするワークフローサーバと、を備えたことを特徴とするワークフローシステム。

【請求項5】 前記設計用コンピュータ端末により設計されたワークフローにおけるプロジェクト変数は、前記ノードと当該ノードの処理順を決定するためのパスとを定めたワークフロー定義および当該ワークフロー定義で参照されるデータ定義とを用いて指定されることを特徴とする請求項4記載のワークフローシステム。

【請求項6】 前記プロジェクト変数は、自己を識別するための定義IDを属性として定義され、前記ワークフローサーバは、定義された前記定義IDを用いて前記プロジェクト変数に関する情報を記憶することを特徴とする請求項4記載のワークフローシステム。

【請求項7】 前記操作用コンピュータ端末は、割り当てられた処理の実行後にアクティビティの終了を前記ワークフローサーバに通知し、

前記ワークフローサーバは、前記ワークフローによる全てのアクティビティを終了した後にプロセスを完了することを特徴とする請求項4記載のワークフローシステム。

【請求項8】 ワークフローを実行する複数のコンピュータ端末に対してネットワークを介して接続される情報処理装置であって、

複数のノードとそれらを結ぶパスとで構成されたワークフロー定義および当該ワークフロー定義で参照されるデータ定義を含めたプロジェクトを管理するプロジェクト管理手段と、

前記プロジェクト管理手段により管理される前記プロジェクトを用いて前記ワークフローの業務単位であるプロセスを管理するプロセス管理手段と、

前記プロセス管理手段により管理される前記プロセスに基づいて前記コンピュータ端末に対してアクセスするユーザ管理手段と、を備えたことを特徴とする情報処理装置。

【請求項9】 前記プロジェクト管理手段は、前記プロジェクトを一意に定める定義IDを用いて当該プロジェクトを管理することを特徴とする請求項8記載の情報処理装置。

【請求項10】 ネットワークに接続された複数のコンピュータ端末によるワークフローを定義する情報処理装置であって、

ワークフローの実行担当者に対して割り当てられる業務であるノードを選定し、選定される当該ノードの処理順を決めるパスをつないでワークフロー定義を決定するワークフロー定義決定手段と、

前記ワークフロー定義決定手段により選定された前記ノードの実行ユーザを定義すると共に、前記パスが有効となる条件を定義するデータ定義手段と、

前記ワークフロー定義決定手段により決定されたワークフロー定義および前記データ定義手段により定義されたデータ定義に対し、これらを特定できる情報を付加して登録させる登録指示手段と、を備えたことを特徴とする情報処理装置。

【請求項11】 前記登録指示手段は、前記ワークフロー定義と前記データ定義とをまとめた単位であるプロジェクトに対して、当該プロジェクトを一意に特定できる定義IDを付加して登録させることを特徴とする請求項10記載の情報処理装置。

【請求項12】 ネットワークに接続された複数のコンピュータ端末によるワークフローを定義するためのワークフロー定義方法であって、

ワークフローの実行担当者に対して割り当てられる業務であるノードを選定し、

選定される前記ノードの1つとしてプロセスを多重化するためのプロジェクト配列型ノードを選定し、

選定される前記ノードおよび選定される前記プロジェク

ト配列型ノードを経路を用いて連結し、選定される前記プロジェクト配列型ノードに配列されるプロジェクトを選定することを特徴とするワークフロー定義方法。

【請求項13】 選定される前記プロジェクトは、配列の個々の要素が異なる種類のプロジェクトであることを特徴とする請求項12記載のワークフロー定義方法。

【請求項14】 前記プロジェクトは、複数のノードとそれらを結ぶパスで構成されたワークフロー定義、およびワークフローの業務単位であるプロセスで設定および参照できるデータの定義であるデータ定義をまとめた単位であることを特徴とする請求項12記載のワークフロー定義方法。

【請求項15】 ネットワークに接続された複数のコンピュータ端末にて実行されるワークフロー定義方法であって、データセットと経路とを統一して扱えるプロジェクト型変数からなるノードをワークフローの1つのノードとして導入し、前記プロジェクト型変数を配列として指定することで、同期的に分岐される複数の子プロセスを1つのノードとして扱うことを特徴とするワークフロー定義方法。

【請求項16】 前記プロジェクト型変数に、明示的な設定がない場合に適用されるプロトタイプを設定することを特徴とする請求項15記載のワークフロー定義方法。

【請求項17】 コンピュータに実行させるプログラムを当該コンピュータの入力手段が読取可能に記憶した記憶媒体であって、前記プログラムは、ワークフローの実行担当者に対して割り当てられる業務であるノードを選定する処理と、選定される前記ノードの1つとしてプロセスを多重化するためのプロジェクト配列型ノードを選定する処理と、前記ノードおよび前記プロジェクト配列型ノードを経路を用いて接続する処理と、前記プロジェクト配列型ノードに配列されるプロジェクトを選定する処理と、を前記コンピュータに実行させることを特徴とする記憶媒体。

【請求項18】 コンピュータに実行させるプログラムを当該コンピュータの入力手段が読取可能に記憶した記憶媒体であって、前記プログラムは、データセットと経路とを統一して扱えるプロジェクト型変数からなるノードをワークフローの1つのノードとして導入する処理と、前記プロジェクト型変数を配列として指定することで、同期的に分岐される複数の子プロセスを1つのノードとして扱う処理と、を前記コンピュータに実行させることを特徴とする記憶媒体。

【請求項19】 コンピュータに、ワークフローの実行担当者に対して割り当てられる業務であるノードを選定する処理と、選定される当該ノードの1つとしてプロセスを多重化するためのプロジェクト配列型ノードを選定する処理と、当該ノードおよび当該プロジェクト配列型ノードを経路を用いて接続する処理と、当該プロジェクト配列型ノードに配列されるプロジェクトを選定する処理とを実行させるプログラムを記憶する記憶手段と、前記記憶手段から前記プログラムを読み出して当該プログラムを送信する送信手段とを備えるプログラム伝送装置。

【請求項20】 コンピュータに、データセットと経路とを統一して扱えるプロジェクト型変数からなるノードをワークフローの1つのノードとして導入する処理と、前記プロジェクト型変数を配列として指定することで、同期的に分岐される複数の子プロセスを1つのノードとして扱う処理とを実行させるプログラムを記憶する記憶手段と、前記記憶手段から前記プログラムを読み出して当該プログラムを送信する送信手段とを備えるプログラム伝送装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、オフィスにおけるワークフローの改善に係り、特に、動的に経路を変更できるワークフローシステム等に関する。

【0002】

【従来の技術】 オフィスにおける事務処理の効率化、ファイルキャビネットのスペースコストの問題、環境問題等から、伝票処理を電子化して、ペーパーレスな帳票処理システムを構築することの必要性が提唱されて久しい。しかしながら、実際の伝票処理は複雑であり、更に例外処理に対する対応の問題などから、ペーパーレスな帳票処理システムは、未だ十分に普及されているとはいえない。また、多数の伝票を電子化するために、伝票定義や経路定義の簡単さが要求され、現実フィットした承認システムとするための柔軟な経路指定の機能が要求されている。

【0003】 ここで、ワークフローシステムに関する従来技術として、例えば、特開平10-49603号公報、特開平10-177610号公報、特開平5-216736号公報が存在する。この特開平10-49603号公報には、異なるプロセス間の回覧、即ち、プロセスAからプロセスBへデータを回覧する場合に、プロセスAで定義されるデータはプロセスBでは読み取ることができないように設定することで、秘密事項の漏洩を防ぐ技術について開示されている。また、特開平10-177610号公報には、複数の担当者における処理を規定して各担当者間でデータを受け渡すワークフローシステムにて、次の担当者に処理を割り付ける前に規定され

たデータ変換を実行することで、各担当者にその業務処理の内容に応じて形式の異なる文書を送る等、業務処理の簡素化を図る技術について示されている。また、特開平5-216736号公報には、電子化された文書(伝票)の処理先が文書や発行者によって異なる場合に、組織を表現するディレクトリを参照することによって、処理すべき文書に依存した経路ルールと送付先を決定する技術について開示されている。

【0004】

【発明が解決しようとする課題】このように、従来のワークフローシステムでは、データの値によって回覧経路を動的に変更するために、個々の担当者をデータ参照により決定するような経路や、固定された外部のワークフロー定義を指定するような経路の定義は可能であった。しかしながら、担当部門を動的に指定するような業務は一般的ではあるものの、それぞれの部門内での回覧経路が部門毎に異なる場合や、その最大回覧部門数を考慮したワークフローを定義する必要があった。

【0005】また、担当者レベルの動的経路や、固定された外部経路と条件分岐を用いたワークフロー定義では、最大回覧部門数が大きくなればなるほど、その経路は複雑で巨大なものとなってしまう、フロー条件の組み合わせを考慮すると、正しい定義をすることさえ困難となる。例えば、ワークフローシステムの一つである電子帳票承認システムにおいて、稟議を回す場合に、多くの部門を並列に回し、全ての部門の承認をもって次の承認に進むケースがよくある。ところが、部門によって承認のパターンは様々な上に、案件によってどの部門に回すかが異なり、例えば、最大、数十部門を回す伝票も現われている。

【0006】更に、従来、文字や数値によるデータを定義し、そのデータの値を参照または設定するようなアクティビティを実現する画面を開発する場合に、専用の開発ツールなどを用いて画面入力項目とデータ名とを関連付ける作業が行われていた。しかしながら、例えば、上述のように数十部門を回すような伝票、および承認経路を設計する場合に、最大数の部門を想定してワークフローを設計する必要がある。そのため、その設計画面が巨大かつ複雑となり、ユーザオペレーションが複雑になるだけでなく、組織変更に対応するための設計変更のコストが多大となっていた。また、指定した担当者によって経路を切り替えるような複雑な動的変更をワークフローに加えるためには、専用のプログラムを開発し、ワークフローの途中で実行させる必要があった。

【0007】本発明は、以上のような技術的課題を解決するためになされたものであって、その目的とするところは、例えば回覧経路が部門毎に異なる複数部門の回覧等、複雑なワークフローを、単純な定義によって実現することにある。また他の目的は、ワークフローの途中で実行させるような専用のプログラムを開発しない場合で

あっても、複雑な経路パターンの組み合わせに対応させることにある。

【0008】

【課題を解決するための手段】かかる目的のもと、本発明は、経路定義において、ノードとしてプロジェクト変数を指定させることで、ダイナミックにサブ経路を変更する仕組みを実現しており、また、プロジェクト変数に配列型の変数を指定させることで、同期タイプの並列経路を1つのノードで多重化して扱うことを可能としている。即ち、本発明は、複数のコンピュータ端末が接続されるネットワークにおいて、ワークフローに従ってコンピュータ端末を利用する担当者間でデータの受け渡しを行うワークフローシステムであって、担当者に割り当てられる処理であるノードとこのノードの処理順を決定するためのパスとを定めたワークフロー定義およびこのワークフロー定義で参照されるデータ定義からなるプロジェクトを作成してワークフローを設計する設計手段と、作成されたプロジェクトを記憶する記憶手段と、記憶されたプロジェクトを用いてワークフローの業務単位であるプロセスを作成して担当者に対して処理を割り当てる管理手段とを備えたことを特徴としている。

【0009】ここで、このデータ定義では、データとしてプロジェクト型およびその配列が定義されることを特徴としている。また、この設計手段は、配列の個々の要素が異なる種類のプロジェクトを扱ってワークフローを設計することを特徴とすれば、複雑な定義からなるワークフローを、プロジェクトを用いた極めてシンプルな定義によって提供できる点で好ましい。

【0010】他の観点から捉えると、本発明が適用されるワークフローシステムでは、ネットワークに接続され、担当者の間でワークフローを実行するための操作用コンピュータ端末と、扱われる処理の単位であるノードに対して複数のプロセスを多重化するためのプロジェクト変数を指定してワークフローを設計する設計用コンピュータ端末と、設計されたワークフローを管理すると共に、ノードに割り当てられた処理であるアクティビティに対応して操作用コンピュータ端末にアクセスするワークフローサーバとを備えたことを特徴している。ここで「プロセス」とは、ワークフローの業務単位であり、例えば、「8月業務報告」と「9月業務報告」とは異なるプロセスであるが、同じ「業務報告」というプロジェクトとなる。従って、この場合、経路パターンは同じであるが、データの内容が異なってくる。これは、同じプロジェクトで異なるプロセスの例だが、ノードで多重化されるプロセスは、異なるプロジェクトの異なるプロセスであっても良い。

【0011】ここで、このプロジェクト変数は、ノードとこのノードの処理順を決定するためのパスとを定めたワークフロー定義およびこのワークフロー定義で参照されるデータ定義とを用いて指定されることを特徴とすれ

ば、例えば、経路やフィールド定義といったリソースを決定する際に有用となる。また、このプロジェクト変数は、自己を識別するための定義IDを属性として定義され、このワークフローサーバは、この定義IDを用いてプロジェクト変数に関する情報を記憶することを特徴とすれば、プロジェクトで別のプロジェクトの経路やフィールドといったリソースを再利用する際にも、プロジェクトを一意に選定できる点で好ましい。更に、操作用コンピュータ端末は、割り当てられた処理の実行後にアクティビティの終了をワークフローサーバに通知し、ワークフローサーバは、ワークフローによる全てのアクティビティを終了した後にプロセスを完了することを特徴としている。

【0012】一方、本発明は、ワークフローを実行する複数のコンピュータ端末に対してネットワークを介して接続されるワークフローサーバ等の情報処理装置であって、複数のノードとそれらを結ぶパスとで構成されたワークフロー定義およびこのワークフロー定義で参照されるデータ定義を含めたプロジェクトを、このプロジェクトを一意に定める定義IDを用いて管理するプロジェクト管理手段と、管理されるこのプロジェクトを用いてワークフローの業務単位であるプロセスを管理するプロセス管理手段と、管理されるプロセスに基づいてコンピュータ端末に対してアクセスするユーザ管理手段とを備えたことを特徴としている。

【0013】また、本発明は、ネットワークに接続された複数のコンピュータ端末によるワークフローを定義するコンピュータ端末等からなる情報処理装置であって、ワークフローの実行担当者に対して割り当てられる業務であるノードを選定し、ノードの処理順を決めるパスをつないでワークフロー定義を決定するワークフロー定義決定手段と、選定されたノードの実行ユーザを定義すると共に、パスが有効となる条件を定義するデータ定義手段と、決定されたワークフロー定義および定義されたデータ定義に対し、これらを特定できる情報として、例えば、ワークフロー定義とデータ定義とをまとめた単位であるプロジェクトに対して、これを一意に特定できる定義IDを付加して登録させる登録指示手段とを備えたことを特徴としている。

【0014】更に、本発明は、ネットワークに接続された複数のコンピュータ端末によるワークフローを定義するためのワークフロー定義方法であって、ワークフローの実行担当者に対して割り当てられる業務であるノードを選定し、選定されるノードの1つとしてプロセス、経路パターンおよびデータセットを多重化するためのプロジェクト配列型ノードを選定し、選定されるノードおよび選定されるプロジェクト配列型ノードを経路を用いて連結し、選定されるプロジェクト配列型ノードに、例えば、配列されるプロジェクトを選定することを特徴としている。

【0015】ここで、このプロジェクトは、複数のノードとそれらを結ぶパスで構成されたワークフロー定義、およびワークフローの業務単位であるプロセスで設定および参照できるデータの定義であるデータ定義をまとめた単位であることを特徴とすることができる。また、この選定されるプロジェクトは、配列の個々の要素が異なる種類のプロジェクトであることを特徴とすれば、配列型の変数を指定させることで、並列経路を1つのノードで多重化することができる点で好ましい。

【0016】他の観点から捉えると、本発明は、ネットワークに接続された複数のコンピュータ端末にて実行されるワークフロー定義方法であって、データセットと経路とを統一して扱えるプロジェクト型変数からなるノードをワークフローの1つのノードとして導入し、プロジェクト型変数を配列として指定することで、同期的に分岐される複数の子プロセス(サブプロセス)を1つのノードとして扱うことを特徴としている。ここで、このプロジェクト型変数には、明示的な設定がない場合に適用されるプロトタイプを設定することを特徴とすることができる。また、ここで用いられる「子プロセス(サブプロセス)」は、例えば、経路パターンとデータセットの両方を意味するものとして扱うことができる。

【0017】一方、本発明は、コンピュータに実行させるプログラムをコンピュータの入力手段が読取可能に記憶した記憶媒体であって、このプログラムは、ワークフローの実行担当者に対して割り当てられる業務であるノードを選定する処理と、このノードの1つとしてプロセス、経路パターンおよびデータセットを多重化するためのプロジェクト配列型ノードを選定する処理と、このノードおよびプロジェクト配列型ノードを経路を用いて接続する処理と、プロジェクト配列型ノードに配列されるプロジェクトを選定する処理とをコンピュータに実行させることを特徴としている。

【0018】また、他の記憶媒体に格納されるプログラムでは、データセットと経路とを統一して扱えるプロジェクト型変数からなるノードをワークフローの1つのノードとして導入する処理と、プロジェクト型変数を配列として指定することで、同期的に分岐される複数の子プロセスを1つのノードとして扱う処理とをコンピュータに実行させることを特徴としている。これらの記憶媒体としては、例えば、CD-ROM等の媒体が該当する。

【0019】更に、本発明が適用されるプログラム伝送装置は、コンピュータに、ワークフローの実行担当者に対して割り当てられる業務であるノードを選定する処理と、選定されるノードの1つとしてプロセス、経路パターンおよびデータセットを多重化するためのプロジェクト配列型ノードを選定する処理と、ノードおよびプロジェクト配列型ノードを経路を用いて接続する処理と、プロジェクト配列型ノードに配列されるプロジェクトを選定する処理とを実行させるプログラムを記憶する記憶手

段と、この記憶手段からプログラムを読み出してプログラムを送信する送信手段とを備えている。

【0020】更にまた、本発明が適用されるプログラム伝送装置は、コンピュータに、データセットと経路とを統一して扱えるプロジェクト型変数からなるノードをワークフローの1つのノードとして導入する処理と、プロジェクト型変数を配列として指定することで、同期的に分岐される複数の子プロセスを1つのノードとして扱う処理とを実行させるプログラムを記憶する記憶手段と、この記憶手段からプログラムを読み出して送信する送信手段とを備えることを特徴としている。

【0021】

【発明の実施の形態】以下、添付図面に示す実施の形態に基づいてこの発明を詳細に説明する。まず、具体的な説明に入る前に、本実施の形態にて用いられる主要な用語について以下に列挙して説明する。

・ワークフロー定義

複数のノードとそれらを結ぶパスで構成された定義。

・ノード

ワークフロー定義で扱う処理定義の単位。実際の処理者とデータ入力画面が割り当てられる。

・データ定義

プロセスで設定および参照できるデータの定義。データの名称、型、配列かどうかを定義する。型には、文字列型や数値型がある。

・データ属性定義

データが保持する値を制限する属性の定義で、データ定義の一部である。文字型データに対する属性には文字数の制限などが、数値型データに対する属性には範囲指定などがある。

・プロジェクト

ワークフロー定義とデータ定義をまとめた単位。プロセス定義と等価である。プロジェクトは定義IDで識別され、プロジェクトが決定されると、そのワークフロー定義とデータ定義も決定される。

・プロセス

ワークフローの業務単位。プロジェクトを具現化したもの。プロセスは、その元となるプロジェクトのワークフロー定義で示される経路パターンに沿って業務が流れる。また、データ定義で示されるデータセットは、個々のプロセスで異なる内容を記憶する。

・アクティビティ

ノードに割り当てられた処理。処理者が割り当てられていれば、担当者の処理になる。

【0022】次に、本実施の形態における具体的な説明に入る。本実施の形態の理解を容易にするために、まず、適用事例を用いて説明する。図18(a)、(b)は、ワークフロー設計者および伝票起票者による操作の一例を示した図であり、本実施の形態が適用される単純な例を示している。同じ経路パターンの多重化の場合、図1

8(a)に示すように、ワークフローの設計者は、あるノードの設定としてプロジェクト型の配列変数PRJ[]を参照するように指定する。ここに、PRJのプロトタイプ属性(明示的な設定がない場合に採用されるもの)として、プロジェクトの定義ID“bumon_prj”を設定する。図18(a)に示される「Project=Field(ID)+サブ経路:bumon」の表現では、「プロジェクト型データを参照するノードでは、そのアクティビティとして、参照されたプロジェクトの定義を使って子プロセスを生成・実行している。また、同じbumonというプロジェクトを使った子プロセスを生成・実行する場合でも、各子プロセスのIDという名前のデータを別々に設定することで、異なるユーザ(もしくは部門)に回覧できる。」という意味を示している。

【0023】図18(b)に示すように、伝票を起票するユーザは、その時点にてダイナミックに承認すべき部門のID(soumu, keiri, eigyou)を入力する。図18(b)に示すINPUTタグによって、これらは変数PRJの各配列要素における変数IDに関連付けられている。データ値が代入される場合、プロジェクトのプロトタイプ属性としてbumon_prjが設定されているので、このノードのアクティビティとしてプロジェクト定義ID(bumon_prj)のプロジェクトを使って作成された子プロセスが設定され、この子プロセス内のデータである変数IDに値が設定される。業務処理として、伝票が処理されるとき、このノードの設定はプロジェクト型の配列を参照するように設定されていることから、配列要素の数だけ子プロセスが生成され、並列に処理される。経路は、プロジェクトbumon_prjで定義されたものが利用され、先頭のノード設定がデータ参照指定でIDという名前のデータを参照するように設定されているので、それぞれID=“soumu”、ID=“keiri”、ID=“eigyou”が評価される。そして、それぞれの部門に送付され、その部門での担当者が承認をする。これによって、一連のワークフローを実現することが可能となる。

【0024】一方、異なる経路パターンの多重化の場合、ワークフロー設計者は、同様に、あるノードの設定としてプロジェクト型の配列変数PRJ[]を参照するように指定する。次に、伝票を起票するユーザは、その時点でダイナミックに承認すべき部門用のプロジェクトの定義ID(soumu_prj, keiri_prj, eigyou_prj)を入力する。INPUTタグによって、これらは変数PRJの各配列要素に別々のプロジェクトとして設定される。この伝票が処理されるとき、このノードではプロジェクト型の配列を参照するように設定されていることから、配列要素の数だけ子プロセスが生成され、並列に処理される。このとき、それぞれ異なるプロジェクトID(soumu_prj, keiri_prj, eigyou_prj)が評価され、異なる種類の子プロセスが設定される。経路は、それぞれのプロジェクトで定義されたものが利用され、別々の経路を用いて

それぞれの部門に送付されて、その部門での担当者が承認する。各子プロセスは、同期されて次のノードに進む。このようにして、異なる経路パターンを多重化したワークフローが実現される。尚、ここで示したフローの本質的な経路設計を含む親プロジェクトは、同じ経路パターンの多重化の場合と異なる経路パターンの多重化の場合で同じとすることが可能である。

【0025】このように、本実施の形態では、ワークフロー定義やデータ定義、表示形式、関連プログラムなどの複数のリソースから構成されるアプリケーションの構成単位をプロジェクトと呼び、このプロジェクトという概念において、ワークフロー定義内のノードなどの要素やデータ定義で定義された変数名といったリソースの名前空間を一意に決定するために、プロジェクト名等からなるプロジェクトIDを採用している。また、このプロジェクトで、別のプロジェクトのワークフロー定義やデータ定義といったリソースを再利用するために、プロジェクト型のデータという概念を導入している。ワークフロー定義上のノードの設定としてこのプロジェクト型データを参照するように指定することで、いわゆる「子プロセス」を実現している。更に、プロジェクト型データを配列型と定義することで、並列な子プロセスを実現している。これらの概念によって、子プロセスや子プロセス内のデータに対して柔軟な制御が可能となる。

【0026】次に、本実施の形態を全体システム構成から順に詳述していく。図1は、本実施の形態が適用されたワークフローシステムの概略構成を示す図である。本実施の形態におけるワークフローシステムは、ワークフローの設計のための設計用コンピュータ端末10、業務を遂行する担当者に対応して設けられたワークフローの操作用コンピュータ端末20、ワークフローを実行するための各種プログラムが格納されるワークフローサーバ30とを備えている。この設計用コンピュータ端末10、操作用コンピュータ端末20、およびワークフローサーバ30は、それぞれ接続されてネットワークを形成している。

【0027】設計用コンピュータ端末10は、ワークフローの設計機能が導入されており、業務処理をワークフローシステムで実現するための定義機能を備えている。業務処理の定義を行うユーザは、この設計用コンピュータ端末10を用いて業務処理手順をワークフローシステムに定義する。操作用コンピュータ端末20は、予め指定された業務を実行する機能があり、複数の業務担当者が操作できるように配置されている。

【0028】設計者は、設計用コンピュータ端末10を用いて、自己の識別情報を入力してワークフローシステムに接続する。また、設計者は、ワークフロー設計機能を用いて、対象業務における処理の流れと処理で用いられるデータ構造とを設計する。図2(a)、(b)は、対象業務における処理の流れを示したプロジェクト(プロセ

ス定義)を示した図であり、図2(a)はワークフロー定義の一例を示し、図2(b)はこのワークフロー定義で参照されるデータ定義の一例を示している。図2(a)において、丸印はノードと呼ばれ、担当者(業務担当者)に割り当てられる業務を示している。矢印はパスと呼ばれ、業務処理の順番、即ち、処理の繋がりを示している。本実施の形態では、全てのノードにて単一のデータ定義が参照され、各処理間で必要となるデータの受け渡しも矢印で示されるパスに従うと仮定している。しかしながら、複数のデータ定義が存在し、異なるデータの受け渡しがパスで定義されていても構わない。図2(b)では、ユーザID(User ID)、名前(Name)として“String(文字列型)”が定義され、アドレス(Address)として“String[]”、Ageとして“Integer(数値型)”が定義されている。

【0029】設計用コンピュータ端末10を用いている設計者は、設計ツールを用いてノードを並べ、処理順を決めるためにパスを繋いでいく。ノードおよびパスに対して、その属性を規定することができる。ノードの属性としては、ノードの名前および担当者となる実行ユーザを定義する。実行ユーザの割り当てタイプには、ユーザIDを直接入力する直接指定、他の処理ユーザとの関係(例えば、ノードA(Node A)の処理ユーザの上司など)で示した関係指定、および、図2(b)で示したようなデータ定義で定義されるデータの値を処理ユーザとするデータ参照指定がある。パスの属性としては、そのパスが有効となる条件を定義する。このようにして設計用コンピュータ端末10にて作成されたプロジェクトは、ワークフローサーバ30に登録される。このとき、プロジェクトを一意に特定することができる定義IDが付加されて登録される。プロジェクトから作成される個々の業務フローがプロセスであり、このプロセスを形成する担当者ごとの処理がアクティビティとなる。

【0030】各担当者は、操作用コンピュータ端末20を用い、自己の識別情報を入力してワークフローシステムに接続する。担当者は、図2(a)に示したようなワークフロー定義を参照して、新たにプロセスを作成したり、自分に割り当てられているアクティビティを表示して、指定されている処理を実行する。各担当者の処理は、図2(a)のワークフロー定義の1つであるノードに対応している。担当者の処理が終了すると、ワークフロー定義のパスに従って次の担当者の処理が割り当てられる。

【0031】図3は、ワークフローサーバ30の概略構成を示した説明図である。本実施の形態におけるワークフローサーバ30は、管理手段として、プロジェクト管理プログラム31、プロセス管理プログラム32、クライアント要求管理プログラム33、およびユーザ管理プログラム34を備えている。また、記憶手段として、ワークフロー定義記憶部35、データ定義記憶部36、プ

ロセス記憶部37、ワークフロー状況記憶部38、およびユーザ情報記憶部39を備えているが、これらの記憶手段はネットワーク内のどこに存在していても構わない。

【0032】プロジェクト管理プログラム31は、設計用コンピュータ端末10にて設計者が定義したプロジェクトのワークフロー定義とデータ定義とに関するデータをワークフロー定義記憶部35およびデータ定義記憶部36に保存して管理する。それぞれのプロジェクトは、定義IDによって識別されている。新たな定義の登録、および既に存在している定義の修正は、全てプロジェクト管理プログラム31を通じて、定義IDを用いて管理する。また、担当者が新規プロセスを作成する際には、その担当者が作成できるプロセスの一覧を、プロジェクト管理プログラム31がワークフロー定義記憶部35とデータ定義記憶部36とに格納されたデータを用いて作成する。

【0033】プロセス管理プログラム32は、プロジェクトの定義を用いて作成されたプロセスの管理を行う。プロセスは、定義IDと、同じ定義から作成された複数のプロセスを識別するためのプロセスIDとを有し、この定義IDとプロセスIDとの組み合わせで管理されている。また、プロセス記憶部37には、現在実行中のアクティビティ、アクティビティを実行している若しくは実行すべき担当者(実行ユーザを含む)、プロセスで用いられているデータが保管されている。これらの情報をもとに、アクティビティの終了時に次の担当者の割り当てが行われる。プロセス管理プログラム32は、プロセスの作成、アクティビティの割り当て、プロセスの完了といったイベントが発生する度に、その状況をワークフロー状況記憶部38に書き込んでいる。

【0034】クライアント要求管理プログラム33は、操作用コンピュータ端末20を用いる担当者の要求を受け付ける機能を有している。また、担当者に現在、割り当てられている(実行ユーザが担当者になっている)アクティビティの一覧を提供したり、担当者のアクティビティ終了要求を受け付ける機能を備えている。また、ワークフロー状況記憶部38に残された情報を用いてプロセスの処理状況を提供している。

【0035】ユーザ管理プログラム34は、ワークフローシステムに参加できる担当者のコントロールを行っている。そのために、ユーザ情報をユーザ情報記憶部39に登録しておき、必要に応じてアクセスしている。新たなユーザが操作用コンピュータ端末20を用いてシステムに接続しようとする場合には、ユーザの入力したIDおよびパスワードのチェックが行われる。また、ユーザ情報記憶部39には、ユーザの属する組織や、組織の階層構造、ユーザの上司関係、ユーザの権限区分等の情報が保存されている。これらの情報は、プロセス管理が次のアクティビティに対して担当者を割り当てる際に参照

される。参照される情報は、例えば、図2(a)に示されるようなワークフロー定義のノードの属性によって異なっている。

【0036】次に、本実施の形態におけるワークフローシステムの動作について説明する。図4(a)、(b)は、登録済みプロジェクトの一例を示した図である。ここでは、プロジェクト「定義A」と、プロジェクト「定義B」とが事前に登録されていると仮定している。それぞれのプロジェクトは、前述したように、ワークフロー定義とデータ定義とをまとめた単位として登録されている。図4(a)に示すプロジェクト「定義A」では、それを用いたプロセスを開始すると、まずノードA(NodeA)に対応したアクティビティの担当者が、プロセスのデータUserIDに代入されている人に割り当てられる。そのアクティビティが終了すると、ノードB(NodeB)に対応したアクティビティが、ノードAのアクティビティを実行したユーザの上司に割り当てられる。

【0037】図4(b)に示すプロジェクト「定義B」において、それを用いたプロセスを開始すると、まずノードA(NodeA)に対応したアクティビティの担当者が、プロセスのデータUserIDに代入されている人に割り当てられる。そのアクティビティが終了すると、ノードC(NodeC)のアクティビティの担当者が、プロセスのデータReviewer(検閲者)に代入されている人に割り当てられる。更に、ノードB(NodeB)に対応したアクティビティの担当者は、ノードAのアクティビティを実行したユーザの上司に割り当てられる。尚、プロジェクト「定義A」は定義IDとして“定義A”を、プロジェクト「定義B」は定義IDとして“定義B”を、それぞれ有している。

【0038】図5は、本実施の形態にて採用されたプロセス定義方法によって作成される定義の一例を示した図である。ここでは、まず、設計者が、図5に示す定義を設計用コンピュータ端末10の設計ツールを用いて作成する。この例では、通常のノード51および通常のノード52との間に、プロジェクト配列型ノード53が設けられている。図5のプロジェクト「定義C」のワークフロー定義におけるノードB(NodeB)と、データ定義におけるDeptsが本実施の形態で採用される定義方法である。ノードBのユーザ割り当ては、従来の定義と同じようにデータ参照となっているが、参照するデータのタイプはプロジェクト型の配列と定義されている。

【0039】このプロジェクト型の配列の要素には他のプロジェクトから作成されたプロセスがその値として代入される。但し、このDeptsはプロトタイプ属性として“定義A”が与えられているため、明示的な定義IDの設定がなければプロジェクト“定義A”のプロセスが代入されることを示している。これらの定義から、このノードBにおけるアクティビティが有効になると、プロジェクト型データDeptsが指すプロセスで置き換えられることを意味し、また配列で定義されている場合は、複数

の別プロセスを同時にこのアクティビティとして開始することを意味している。定義されたプロジェクトはワークフローサーバ30に送られ、プロジェクト管理プログラム31を経てワークフロー定義記憶部35とデータ定義記憶部36とに保存される。ここでは、定義IDとして“定義C”を与えて保存したと仮定する。

【0040】本実施の形態におけるワークフローシステムにて、例えば、担当者Aが新規の業務フローとして図5のプロジェクト「定義C」を用いて新たなプロセスを作成する場合、新規作成要求がワークフローサーバ30のクライアント要求管理プログラム33に送られる。要求を受け取ったクライアント要求管理プログラム33は、プロセス管理プログラム32に新規プロセスの作成を要求する。そして、プロセス記憶部37内に、定義IDとして“定義C”が、プロセスIDとして“C001”が設定され、新たなプロセスとして作成されて保存される。

【0041】図6は、プロセス記憶部37における状態の一例を示した図である。プロセス記憶部37内における個々のプロセスは、定義IDとプロセスIDとの組み合わせで識別されるため、この識別表現として“定義C：C001”のように記述される。また、このプロセスの開始要求も同様にクライアント要求管理プログラム33を経てプロセス管理プログラム32に送られ、アクティビティとして“NodeA”が、実行ユーザとして“担当者A”が設定される。

【0042】図6に示す状態で、担当者Aが操作用コンピュータ端末20を用いてワークフローシステムに接続すると、実行ユーザとして自分に割り当てられた“定義C：C001”におけるアクティビティ“NodeA”を知ることができる。担当者Aが更に端末からこのアクティビティの開始を要求すると、関連付けられたプログラムが起動され、担当者Aは処理画面を得ることができる。

【0043】図7は、担当者Aが得るワークフロー操作の処理画面の一例を示した図であり、図8は、クライアント要求管理プログラム33に送られるデータを示した図である。ここでは、代入先データ名毎に対応する設定値が定められている。図7に示す処理画面には、プロジェクト「定義C」のデータ定義で定義されたデータ構造に対するデータ入力域が設けられている。この図7の例では、ラベル「名前：」および「課題：」で示されるテキスト入力域が、図8に示すように、それぞれプロジェクト「定義C」のデータ構造における代入先データ名“Name”および“Subject”に関連付けられている。また、図7に示す「回覧先：」というラベルで示されたテーブルの1行目の1カラム目が、図8に示す“Depts[1]”に関連付けられ、2カラム目が“Depts[2].User ID”というデータ名に関連付けられている。2行目以降も同様に“Depts[n]”と“Depts[n].User ID”というデータ名に関連付けられている。但し、図5にて説明したように、この

例では、プロトタイプ属性として“定義A”が与えられているため、明示的な定義IDの設定がない“Depts[1]”および“Depts[2]”では、プロジェクト“定義A”のプロセスが代入される。担当者Aがこの操作画面で図7に示すように入力して処理を確定した場合、その情報は、図8に示すような形式でワークフローサーバ30のクライアント要求管理プログラム33に送られる。

【0044】図9は、クライアント要求管理プログラム33にて実行されるデータ設定の評価手順を示すフローチャートである。ここでは、図7で示した入力画面における「回覧先：」の1行目のデータ設定を例にとって説明する。ここでは、予め入力画面で入力項目に割り当てられた代入先データ名は、“Depts[1].User ID”であり、入力された値“総務”が設定値となる。これは図8の3行目に相当する設定項目である。

【0045】まず、クライアント要求管理プログラム33がこの値を設定する要求を受け取ると、現在のプロセス“定義C：C001”を「親プロセス」として評価ロジックが呼び出される(ステップ101)。そして、「評価データ名」に代入先データ名の最初のデータ名部分を設定し(ステップ102)、「INDEX」に代入先データ名における最初のデータ名の配列添え字部分を設定し(ステップ103)、「残りフレーズ」に代入先データ名から評価データ名とINDEXを除いた部分を設定する(ステップ104)。ここでは、代入先データ名が“Depts[1].User ID”であることから、「評価データ名」として“Depts”、「INDEX」として“[1]”、「残りフレーズ」として“User ID”が得られる。

【0046】次に、親プロセスにおいて、評価データ名のデータが定義されているか否かが判断される(ステップ105)。定義されていない場合には、データ定義で定義されていないデータに値を設定しようとしたものとして、エラーとなる(ステップ106)。ステップ105にて定義されている場合には、「評価データ」に親プロセス内で評価データ名を持つデータが設定される(ステップ107)。このプロセスのプロジェクト「定義C」のデータ定義では“Depts”が定義されていることから、その値が「評価データ」とされる。

【0047】次に、評価データの型がプロジェクト型か否かが判断される(ステップ108)。プロジェクト型ではない場合には、評価データのINDEX要素に設定値が代入される(ステップ109)。プロジェクト型である場合には、残りフレーズが空ではないか否かが判断される(ステップ110)。前述した例では、ステップ106の「評価データ」である“Depts”は、「定義C」のデータ定義において“Project[]”と定義されているので、プロジェクト型の配列であることがわかる。また、「残りフレーズ」には“User ID”が設定されているため空ではない。

【0048】ステップ110にて、「残りフレーズ」が空

ではない場合には、「評価データ」がまだプロセスを指していないか否かが判断される(ステップ111)。この例で「評価データ」がプロセスを指しているかどうかを確認すると、図6で“定義C: C001”のデータを見れば、この状態ではまだプロセスは指していないことが解かるので、プロトタイプ属性の確認に移る。即ち、評価データにプロトタイプ属性が定義されているか否かが判断される(ステップ112)。定義されていない場合には、プロトタイプ属性が定義されていないデータの内部データに対し、定義IDを設定しないで直接、アクセスしようとしたものとして、エラーとなる(ステップ115)。この例では、「評価データ」である“Depts”は、「定義C」のデータ定義においてプロトタイプ属性が“定義A”と定義されている。そこで、定義IDにプロトタイプ属性を設定し、プロセスを作成し、また、評価データのINDEX要素が作成したプロセスを指すように設定する(ステップ113)。即ち、定義IDが“定義A”であるプロジェクトからプロセスが作成され、プロセス記憶部37の中に、“定義A”および“A001”を識別子として有するプロセスが保存される。また、「評価データ」の「INDEX」要素である“Depts[1]”には、そのプロセスを指すように“定義A: A001”が設定される。

【0049】次に、親プロセスに「評価データ」の「INDEX」要素が指すプロセスを設定し、代入先データ名に残りのフレーズを設定する(ステップ114)。この例では、「親プロセス」を「評価データ」の「INDEX」要素“Depts[1]”が指すプロセス“定義A: A001”に設定し、「代入先データ名」に「残りフレーズ」“UserID”を設定する。その後、ステップ102にジャンプする。ステップ111にて、評価データがプロセスを指している場合には、直接、ステップ114へ移行する。

【0050】更に評価を進めると、「評価データ名」として“UserID”が得られ(ステップ102)、「INDEX」として“ ”(空文字)が得られ(ステップ103)、「残りフレーズ」として“ ”(空文字)が得られる(ステップ104)。このプロセスのプロジェクト「定義A」のデータ定義では“UserID”が定義されているので(ステップ105)、その値を「評価データ」とする(ステップ107)。この「評価データ」の「INDEX」要素である“UserID”は、「定義A」のデータ定義において“String”と定義されているので文字型であってプロジェクト型ではないことがわかる(ステップ108)。したがって、最終的に「親プロセス」か否かを示すプロセス“定義A: A001”のデータ“UserID”に、設定値“総務”が設定される(ステップ109)。

【0051】ステップ110にて、残りフレーズが空である場合には、「評価データ」がまだプロセスを指していないか否かが判断され(ステップ116)、指していない場合には、定義IDに設定値を設定し、プロセスを作成し、「評価データ」の「INDEX」要素が作成したプロセスを

指すように設定する(ステップ117)。ステップ116にてプロセスを指している場合には、「評価データ」の定義IDが設定値と等しい場合には(ステップ118)、既に設定されている定義IDとは異なる定義IDを設定しようとしたものとしてエラーとなる。

【0052】図10は、図8に示される全てのデータを評価した結果として、プロセス記憶部37にて得られる状態の一例を示した図である。ここでは、プロジェクトを識別するための定義ID、プロセスID、親プロセス、アクティビティ、実行ユーザ、データ、子プロセス数の各項目に対して、対応する評価結果を得ることができる。例えば、担当者Aがアクティビティの処理を終了したとき、終了要求がワークフローサーバ30のクライアント要求管理プログラム33に送られる。クライアント要求管理プログラム33は、プロセス管理プログラム32を呼び出し、“定義C: C001”のプロセスをプロセス記憶部37から取り出し、プロジェクト「定義C」のワークフロー定義をワークフロー定義記憶部35から取り出してフロー評価を行う。

【0053】図11は、本実施の形態におけるフロー評価手順を示したフローチャートである。このフロー評価手順では、図10に示すようなプロセス記録部37の状態から、まず、子プロセス数が0か否かが判断され(ステップ201)、0でない場合には、評価手順の処理は終了する。子プロセス数が0である場合には、アクティビティ名を取得する(ステップ202)。ここで、アクティビティ名が空か否かが判断され(ステップ203)、空でない場合には、アクティビティ名を起点とするパスをワークフロー定義記憶部35から検索する(ステップ204)。

【0054】次に、アクティビティ名を起点とするパスが存在しているか否かが判断され(ステップ205)、存在していない場合には、完了処理が実行されて(ステップ206)、処理が終了する。アクティビティ名を起点とするパスが存在する場合には、パスの終点ノードに指定されているノードをフロー定義から検索し(ステップ207)、検索されたノードの指定内容により実行ユーザIDを特定する(ステップ208)。ステップ203でアクティビティ名が空である場合には、ワークフロー定義記憶部35から先頭になるアクティビティ名を検索し(ステップ209)、ステップ208に移行する。このステップ208の後に、ノードをプロセスのアクティビティに、特定したユーザIDを実行ユーザIDにセットする(ステップ210)。そして、マッチした経路がまだあるか否かが判断され(ステップ211)、まだある場合には、プロセスを複製して(ステップ212)、ステップ207に戻り、マッチした経路がない場合には、フロー評価手順の処理が終了する。

【0055】図12は、本実施の形態における実行ユーザ割り当て手順を示したフローチャートである。まず、

割り当てタイプを取得し(ステップ301)、直接指定か否かが判断される(ステップ302)。直接指定である場合には、定義で指定されたユーザを取り出し(ステップ303)、実行ユーザ割り当て手順の処理が終了する。直接指定でない場合には、関係指定か否かで手順が異なる(ステップ304)。関係指定である場合には、指定された関係を取り出し(ステップ305)、関係を使ってユーザ情報を検索し(ステップ306)、関係を満たすユーザを取得して(ステップ307)、実行ユーザ割り当て手順の処理が終了する。

【0056】ステップ304で関係指定でない場合には、データ参照か否かが判断され(ステップ308)、データ参照でない場合には、実行ユーザ割り当て手順の処理が終了する。データ参照である場合には、指定されたデータの型をチェックするが(ステップ309)、データの型がプロジェクト型であるか否かでその後の処理が異なる(ステップ310)。即ち、プロジェクト型である場合には、データで示される子プロセスを取得し(ステップ311)、子プロセス数にデータの要素数(開始する子プロセスの数)を代入する(ステップ312)。そして、全ての子プロセスに対してフロー評価を行ってプロセスを開始して(ステップ313)、一連の処理が終了する。一方、プロジェクト型でない場合には、データ値を取り出し(ステップ314)、データ値をユーザとして(ステップ315)、実行ユーザ割り当て手順の処理が終了する。

【0057】図13は、本実施の形態における完了処理の手順を示したフローチャートである。完了処理では、まず、プロセスのアクティビティをクリアし(ステップ401)、実行ユーザをクリアする(ステップ402)。次に、親プロセスを持つ子プロセスかどうかチェックされ(ステップ403、ステップ404)、子プロセスである場合には、親プロセスの子プロセスカウントを1つ減らし(ステップ405)、親プロセスのフロー評価を実行する(ステップ406)。最後に、プロセス記憶部37から情報を削除して(ステップ407)、完了処理が終了する。子プロセスでない場合には、そのままプロセス記憶部37から情報を削除して(ステップ407)、完了処理が終了する。

【0058】図14は、図11、図12、図13に示した処理を実行することによるプロセス記憶部37の状態を示した図である。図11～図13の処理によって、図10の状態が図14に示すように変更されている。この図14に示す状態では、子プロセス“定義A:A001”が“総務”というユーザIDを持った担当者に、子プロセス“定義A:A002”が“経理”というユーザIDを持った担当者に、子プロセス“定義B:C001”が“営業”というユーザIDを持った担当者に、それぞれ割り当てられている。即ち、3つの子プロセスが並列に実行されている。割り当てられたユーザが本実施

の形態におけるワークフローシステムに接続すると、自分が実行ユーザになっているアクティビティが存在することが分かり、それを実行することが可能となる。また、例えば、“総務”というユーザIDを持った担当者がそのアクティビティを終了すると、プロセス“定義A:A001”は、その上司に割り当てられる。

【0059】図15は、今までの例における全体のプロセスフローを示した図である。このプロセスフローでは、図5にて示したプロジェクト「定義C」において、ノードB(NodeB)のアクティビティが有効になり、プロジェクト型データDepts[1]～[3]が指すプロセスで置き換えられている。即ち、定義Aの2つの経路と定義Bの経路とからなる並列経路が、1つのノードで多重化して扱われており、通常のノード51と52との間に、3つのサブ経路からなるノード類54を実行させることができる。

【0060】図16は、図7に示した「回覧先:」のデータとして、他のデータを指定した場合を示している。図8にて説明した場合と同様に、この図16に示されるデータは、クライアント要求管理プログラム33に送られる。ここでは、Depts[1]からDepts[4]まで、プロジェクト配列型のデータとして扱われる。また、Depts[1]およびDepts[2]は“定義B”、Depts[3]およびDepts[4]は“定義A”と設定され、Depts[1].UserIDからDepts[4].UserIDまで、User1～User4が設定されている。

【0061】図17は、「回覧先:」として図16を指定した場合の全体のプロセスフローを示した図である。ここでは、図5にて示したプロジェクト「定義C」において、ノードB(NodeB)のアクティビティが有効になり、プロジェクト型データDepts[1]～[4]が指すプロセスで置き換えられている。即ち、定義Aの2つの経路と定義Bの2つ経路とからなる並列経路が、1つのノードで多重化して扱われており、通常のノード51と52との間に、4つのサブ経路からなるノード類55を実行させることができる。

【0062】以上、本実施の形態における処理をまとめると、本実施の形態では、データ定義で定義できるデータとして、プロジェクト型およびその配列を定義できるようにしている。また、プロジェクト型データは、属性として、プロトタイプ属性と呼ばれる、事前指定プロジェクトの定義IDを指定することができる。プロセスを作成すると、そのプロセスが定義されているプロジェクトのデータ定義に従ってデータセットが作成されプロセス記憶部37に記憶される。このようにして作成されたデータには、各ノードに関連付けられた画面を通して値を設定することができる。

【0063】プロジェクト型として定義されたデータには前述したように定義IDが添えられている。プロジェクトの定義IDを指定すると、そのデータには指定したプロジェクトのデータ定義に従った新たなデータセット

が作成され、その後はその新たなデータを設定および参照することができる。特にプロジェクトの定義IDを指定しなくても、プロトタイプ属性に定義IDが設定されている場合は、そのプロトタイプ属性であるプロジェクトのデータ定義に従ったデータを初めて参照または設定した時点で自動的にデータセットを作成することが可能となる。従って、明示的に定義IDを指定する必要はない。また、配列データの場合には、必要な要素数だけプロジェクトの定義IDを指定することで、1つのデータで複数のプロジェクトのデータを扱うことが可能となる。この配列データにおいてもプロトタイプ属性が設定されている場合は、値を初めて参照・設定した時点で、その配列要素に対して自動的にデータセットが作成される。

【0064】プロセス作成直後、もしくはアクティビティが終了すると、ワークフロー定義に従って次に有効となるノードを見つけ出す。この有効となるノードに個別の担当者が割り当てられている場合には、その担当者に処理を割り当てる。また、ノードにデータを参照するよう定義されていた場合は、そのデータが指し示す値を使って担当者を識別し、その担当者が処理を割り当てる。ただし、そのノードがプロジェクト型のデータを参照するよう定義されている場合は、そのデータが指し示す別のワークフロー定義から子プロセスを作成し、この子プロセス自体をこのノードのアクティビティとする。このとき、実際の処理担当者は、子プロセスの先頭ノードを評価して決定することが可能となる。更に、そのノードがプロジェクト型のデータを配列として記憶している場合には、その配列の要素すべてに対して子プロセスを作成し、その子プロセス自体をこのノードのアクティビティとすることができる。この配列を参照して作成された子プロセス群は、そのすべての子プロセスの完了をもって、親プロセスにおけるノードのアクティビティの終了とされる。

【0065】このように、本実施の形態によれば、経路定義において、ノードとしてプロジェクト変数を指定させることで、ダイナミックにサブ経路を変更することが可能となる。また、そのプロジェクト変数に配列型の変数を指定させることで、同期タイプの並列経路を1つのノードで多重化して扱うことができる。言い換えれば、本質的な経路定義では、ノードとして唯一プロジェクト変数を指定するだけなので、ユーザオペレーションを極めてシンプルにすることができ、組織変更などに簡単に対応することが可能となる。また、担当者である起票者が伝票上で承認先部門を指定するだけで、複数部門に並列に承認依頼ができる稟議用の伝票等の設計が容易となる。

【0066】尚、本実施の形態にて説明したワークフローを実行するプログラムは、記憶媒体に格納されて販売等に供されるが、この記憶媒体としては、これらの処理

を実行するためのソフトウェアが格納された、例えばCD-ROM等の媒体が代表的なものである。また、プログラムをネット等を介してダウンロードする態様では、プログラムの伝送装置における媒体や、ダウンロードした後のハードディスク等の記憶媒体も含まれることは言うまでもない。これらの記憶媒体に格納されたプログラムは、入力手段としての、例えばCD-ROMドライバによって読み取られることが可能である。また、プログラムの伝送装置では、例えば、インターネットに接続されたコンピュータ端末からのダウンロード要求に応じて、本実施の形態を実現可能なプログラムを供給できるインターフェイス手段(送信手段)を備えていれば良い。

【0067】

【発明の効果】以上説明したように、本発明によれば、複雑なワークフロー設計を単純な定義によって実現することが可能となる。

【図面の簡単な説明】

【図1】 本実施の形態が適用されたワークフローシステムの概略構成を示す図である。

【図2】 (a)、(b)は、対象業務における処理の流れを示したプロジェクト(プロセス定義)を示した図である。

【図3】 ワークフローサーバ30の概略構成を示した説明図である。

【図4】 (a)、(b)は、登録済みプロジェクトの一例を示した図である。

【図5】 本実施の形態にて採用されたプロセス定義方法によって作成される定義の一例を示した図である。

【図6】 プロセス記憶部37における状態の一例を示した図である。

【図7】 担当者が得るワークフロー操作の処理画面の一例を示した図である。

【図8】 クライアント要求管理プログラム33に送られるデータを示した図である。

【図9】 クライアント要求管理プログラム33にて実行されるデータ設定の評価手順を示すフローチャートである。

【図10】 図8に示される全てのデータを評価した結果として、プロセス記憶部37にて得られる状態の一例を示した図である。

【図11】 本実施の形態におけるフロー評価手順を示したフローチャートである。

【図12】 本実施の形態における実行ユーザ割り当て手順を示したフローチャートである。

【図13】 本実施の形態における完了処理の手順を示したフローチャートである。

【図14】 図11、図12、図13に示した処理を実行することによるプロセス記憶部37の状態を示した図である。

【図15】 全体のプロセスフローを示した図である。

【図16】 図7に示した「回覧先:」のデータとして、他のデータを指定した場合を示した図である。

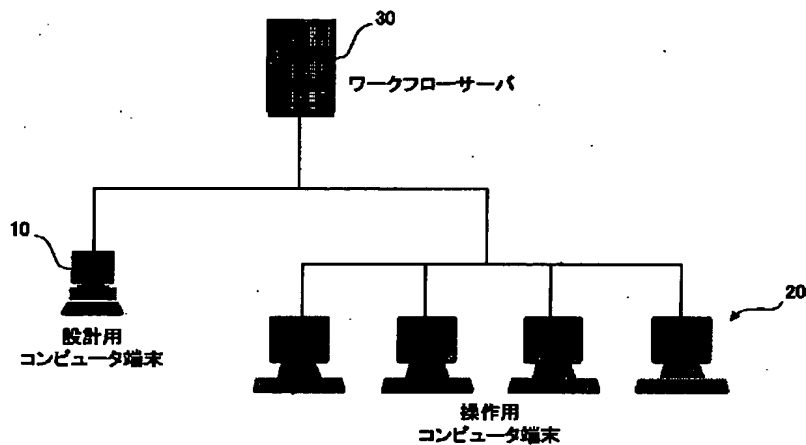
【図17】 「回覧先:」として図16を指定した場合の全体のプロセスフローを示した図である。

【図18】 (a)、(b)は、ワークフロー設計者および伝票起票者による操作の一例を示した図である。

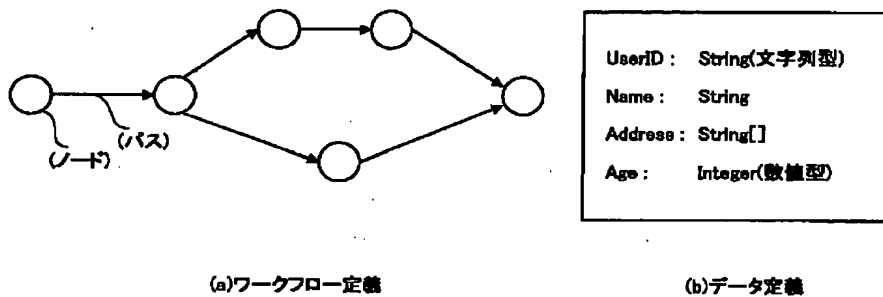
【符号の説明】

10…設計用コンピュータ端末、20…操作用コンピュータ端末、30…ワークフローサーバ、31…プロジェクト管理プログラム、32…プロセス管理プログラム、33…クライアント要求管理プログラム、34…ユーザ管理プログラム、35…ワークフロー定義記憶部、36…データ定義記憶部、37…プロセス記憶部、38…ワークフロー状況記憶部、39…ユーザ情報記憶部

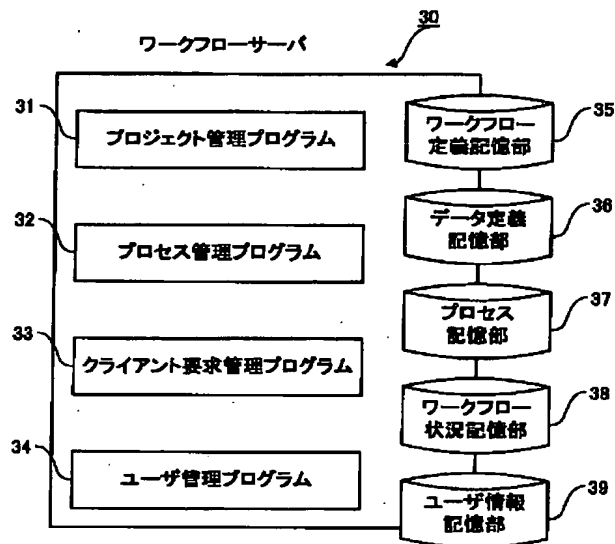
【図1】



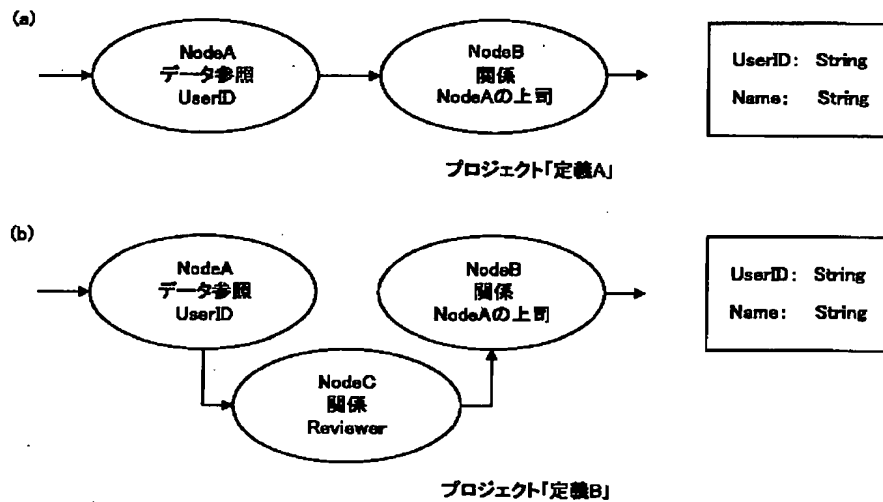
【図2】



【図3】



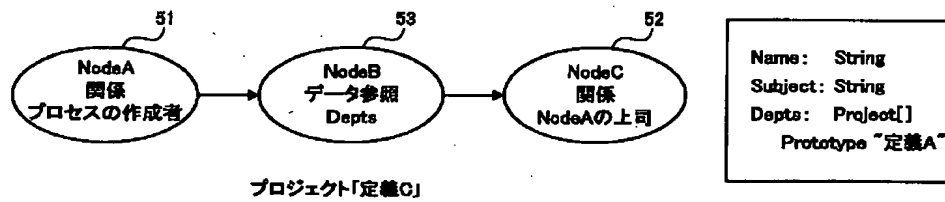
【図4】



【図6】

定義ID	プロセスID	親プロセス	アクティビティ	実行ユーザ	データ	子プロセス数
定義C	C001		NodeA	担当者A	Name = Subject = Depts[] =	

【図5】



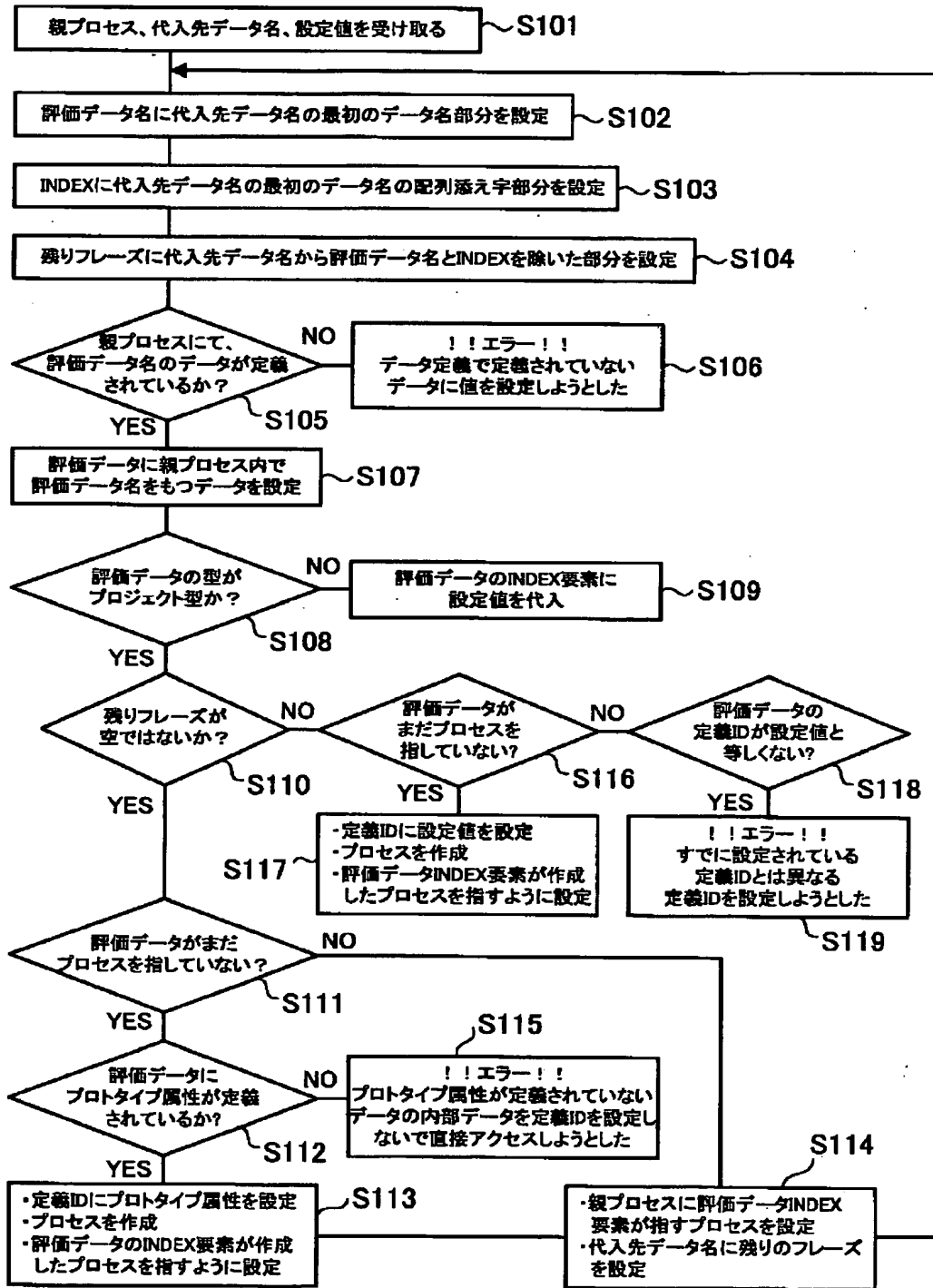
【図7】

名前:	H. Ohsaki
課題:	予算申請
回覧先:	
1	総務
2	経理
3 定義B	営業
4	

【図8】

代入先データ名	設定値
Name	H. Ohsaki
Subject	課題
Depts[1]. UserID	総務
Depts[2]. UserID	経理
Depts[3]	定義B
Depts[3]. UserID	営業

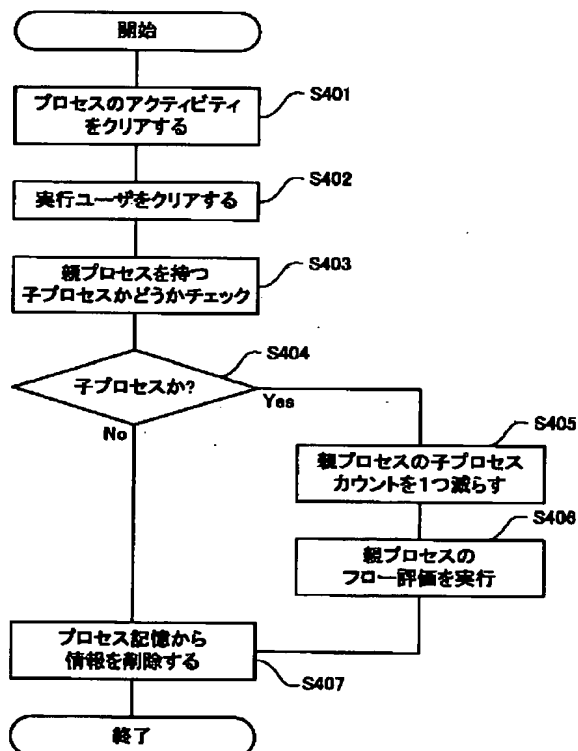
【図9】



【図10】

定義ID	プロセスID	親プロセス	アクティビティ	実行ユーザ	データ	子プロセス数
定義0	0001		NodaA	担当者A	Name="H. Ohsaki" Subject="特許申請" Depts[1]=定義A:A001 Depts[2]=定義A:A002 Depts[3]=定義B:B001	
定義A	A001	定義C:C001			UserID="総務"	
定義A	A002	定義C:C001			UserID="経理"	
定義B	B001	定義C:C001			UserID="営業"	

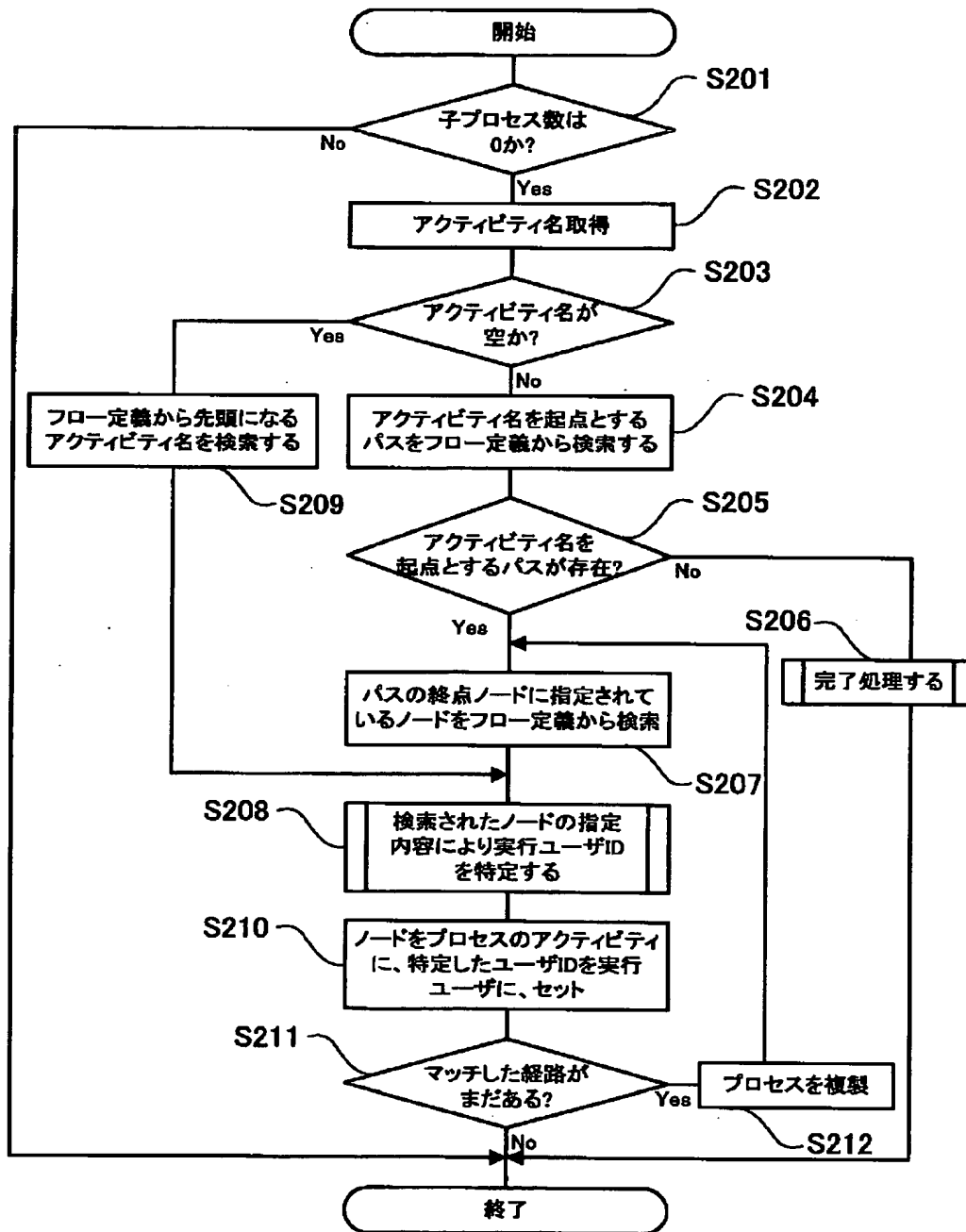
【図13】



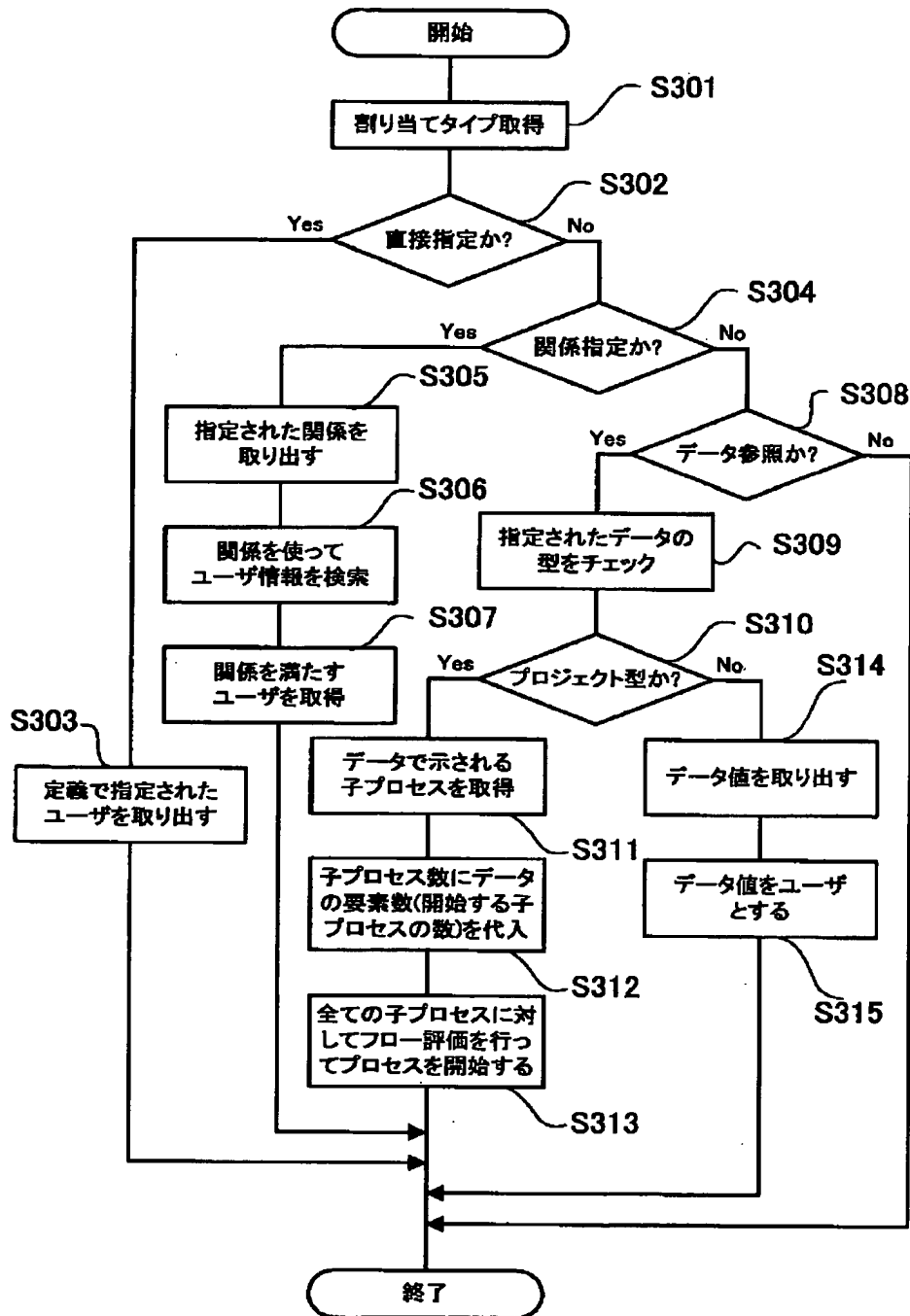
【図16】

代入先データ名	設定値
Depts[1]	定義B
Depts[1]. UserID	User1
Depts[2]	定義B
Depts[2]. UserID	User2
Depts[3]	定義A
Depts[3]. UserID	User3
Depts[4]	定義A
Depts[4]. UserID	User4

【図11】



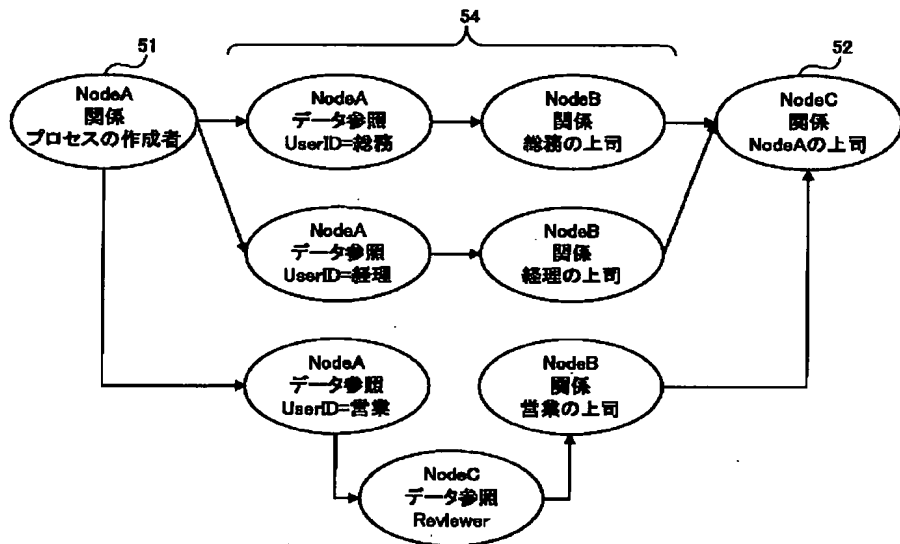
【図12】



【図14】

定義ID	プロセスID	親プロセス	アクティビティ	実行ユーザ	データ	子プロセス数
定義C	C001		NodeB		Name="H. Ohsaki" Subject="特許申請" Depths[1]=定義A:A001 Depths[2]=定義A:A002 Depths[3]=定義B:B001	3
定義A	A001	定義C:C001	NodeA	総務	UserID="総務"	
定義A	A002	定義C:C001	NodeA	経理	UserID="経理"	
定義B	B001	定義C:C001	NodeA	営業	UserID="営業"	

【図15】



フロントページの続き

(72)発明者 津田 和幸

神奈川県大和市下鶴間1623番地14 日本ア
イ・ビー・エム株式会社 大和事業所内

Fターム(参考) 5B049 AA02 BB00 FF07 GG04 GG07